

A typical situation that I am referring to is a ticketing system. These could be Plant Hire, Car Hire Skip Hire or other trade sales that use tickets to record charges.

As previously a developer of numerous systems I used my own standard file system in order to have consistency and to allow reusing as much code as I could. I do not have table names like Customer, ClientForCarHire, SupplierOfConsumables etc. I use a numbering system. I also tend to abbreviate field names and if possible do not exceed 8 characters. Using numbers will sort your Access Objects alphabetically. Also, in code I find it is easier to refer to variables and where they are from using a number to link them. You know that a Query, Form, Report, or Function with 10 in its name relates to the T10 table in the system. It saves developing cryptic and lengthy names that may be confusing in the future. Also, avoid spaces in Filenames, TableNames and FieldNames. However, as you are maybe not creating many different systems it may not be a problem. I would add that if your system has been started, maybe do not switch your naming system later. Otherwise you may enter into a nightmare world of your own making!

As an example for you I presume a Client/Customer table, a Stock table and a Ticket Table. Plus a System Table. These are what may be partial structures.

T90System

NextInv
SetInvNo
PostNo
Other standing data.....etc

T40Stock

Stcode
StDescr
Rate
VATCode
More.... etc

T60Client

ACODE.....this must match the Accounts code in accounting system: 8 long String
CustCode..... this allows a client to have sites or depots etc : 8 long String
Cname
Caddr1
Caddr2
Caddr3
Caddr4
PCODE
Tel, fax etc..maybe invoice by settings

T20Sites

Acode.....link to T60
CustCode..... this allows a client to have sites or depots etc : 8 long String
SAddr1
SAddr2
SAddr3...etc

T10Ticket

CustCode.....links to the T60Client
TicketNo
TicketDate
StCode
StDescr
Quant
Rate
VATCode
TotCharge
TotVAT
InvNo
InvDate
PostNo.....default = 0

Whilst T10 is not fully normalised, it does give flexibility and these days, disk space is not an issue. The T20Sites would be used if depots or sites are required to separate invoices to a client.

NOTE! You can of course link the T40 with T10 and avoid some field duplication. However, only do that if you are totally assured that a Rate, Description, or StockCode in T40 will not change between a ticket being added and it being formed into invoices. Or indeed for a re-print of invoices or analysis later. You have the StCode, StDescr and Rate in T10 on the day the ticket was added. If a description or rate is amended in T40 the change will show on the invoices. If a description is amended to a code a re-print of invoices will show that change. You should also prevent a delete of a stock code if it is in use on invoiced tickets. If you allow someone to cause you problems later, you can be assured they will.

If the system has to have price increases on a day then it is more complex. You will need price lists. You can then fill in the new prices and then at the start of the day switch them with a batch update. If you have many stock codes then it may not be practical to do otherwise. Maybe better to get everything 100% operational before thinking about price updates.

The process is:

Over a month there will be a number of tickets in the T10 table. You then form a query from the T10 table sorted on the CustCode + TicketDate, with a PostNo = 0

First zero all invoice numbers with a zero PostNo. In case you are re-forming

You then open the Query and inside a Do While NOT EOF...Loop add the number until the T10.CustCode changes.

As you proceed you will start at the first customer and calculate the TotCharge and TotVAT and fill in the invoice number.

You will start with the Invoice Number in the T90.NextInv. After the end of one CustCode increment the invoice number and repeat for the next until all are processed. When you are finished an invoice number will have been added to each ticket with a zero post number.

You can then update the last invoice +1 into the T90.NextInv. You can now view, print or send to a PDF the invoices and check them if necessary.

A Query will link the T10Client with T10.Custcode and T60.CustCode, so you can show the full Client details on the formed invoice header. If T20Sites is used in the analysis is used then the T10, T20 and T60 links would be used and invoices would be by Site.

If there are errors or omissions you can simply re-form the invoices. You could reform invoices as often as you like, providing that they have a zero PostNo. Once the invoices are correct you will add the PostNumber to the Tickets that have an Invoice Number and a zero PostNo. Then create a unique Import CSV file with the postnumber as a reference for Sage or other accounting system in use. The import will transfer them to their matching AccountCode in accounts.

Next update T90.NextInv to T90.SetInvNo +1. (or maybe not +1, depending on where you advance it) At the end they should have the same value and holding the next invoice number ready for when you form the next batch of invoices. Tickets with a PostNo that isn't zero cannot be altered. This is because once the accounting system has that batch imported, it would cause chaos and invalidate the system if you allow changes. You can then re-print or export any batch of invoices, as well as recreating the accounts import. You accumulate batches of invoices that you can report on. Yet you can still report or analyse over many or all batches.

It is simple to create a header with a Query of Tickets and charge totals for a single invoice number and display all of the related charges. You can open the same table multiple times.

Once the Ticket file becomes huge you can archive to another table, if it is slowing the application. Alternatively, once posted you could send all posted invoices to an invoiced table and remove them from the T10. Your choice. I would do that if there were huge numbers of tickets, or maybe later if the size gives problems. Initially it may be better just filtering out the Posteds as you have all data in one table and it is easier to view, check and report on.

I will reiterate that you should store the result of all calculations in Large Integer fields. The addition of these will always be the same, which may not be the case with a Single, Double or Currency. It is easy to then correctly place the decimal with string handling on reports. (or other method you choose)